**API –** Application Programming Interface

**CO2 – Dióxido de Carbono**

**REST –** Representational State Transfer

**Resumo**

Com a evolução, a humanidade adaptou-se e produziu invenções que hoje em dia são essenciais na sua maioria. No entanto é possível observar as consequências negativas destas criações no meio ambiente, sendo assim fundamental procurar alternativas para os diversos poluentes do nosso dia a dia, que ameaçam o planeta e consequentemente a vida dos seus habitantes e gerações futuras. Uma das indústrias mais poluentes é a indústria dos transportes, na qual se inserem os transportes privados.

Para aumentar a sustentabilidade (preservação do meio ambiente), custos pessoais, diminuir o congestionamento de tráfego, entre outros, surge o conceito de sistema de boleias, do inglês Carpooling System. Um sistema de boleias, pode ser descrito como um facilitador da partilha de lugares disponíveis num veículo particular que realiza uma determinada viagem.

O objetivo deste projeto denominado de Dynamic Carpooling System é conceber uma solução informática que facilita aos membros da comunidade de Carpooling o acesso a informação em qualquer lugar e em tempo real através de um dispositivo móvel, sobre ofertas de boleias de viagens em curso.

\textbf {Palavras-Chave:} Carpooling, Dinâmico, Mobilidade Sustentável, Sistema de Boleias, Serviço.

**Abstract**

With evolution humanity adapted and produced inventions that nowadays are mostly essential. However, it’s possible to watch the negative consequences of these creations on the environment, therefore it’s crucial to look for alternatives for the different pollutants of our day-to-day life which threatens the planet and consequently the lives of its inhabitants and future generations. One of the most polluting industries is the transport industry, which includes private transport.

To increase sustainability (environment preservation), personal costs, decrease traffic congestion, among others, the concept Carpooling System appears. A Carpooling System can be described as facilitating the sharing available seats on a private vehicle that performs a certain trip.

The objective of this project called Dynamic Carpooling System is to conceive a computer solution that makes it easier to the members of the carpooling community to access information anywhere and in real time through a mobile device, about the offers of rides in progress.

\textbf {Key-Words:} Carpooling, Dynamic, Sustainable Mobility, Service.

**Agradecimentos**

A nós e a nós mesmos

**Introdução**

Cada vez mais somos confrontados com as consequências da evolução e aumento da população que se refletiram no uso excessivo e impróprio dos recursos naturais que o planeta nos fornece. Nos dias de hoje já não imaginamos o nosso o dia a dia sem algumas das invenções que surgiram, sendo uma delas e a que será abordada neste projeto, a criação da indústria automóvel uma das indústrias mais poluentes do mundo.

A grande quantidade de carros nos dias de hoje é originária de muita da poluição atmosférica existente. Este tipo de poluição traduz-se na presença de substâncias poluidoras no ar atmosférico e representa um problema grave para a saúde da população, dado que pode desencadear problemas cardiovasculares e respiratórios. De acordo com a OMS (Organização Mundial de Saúde) todos os anos morrem sete milhões de pessoas por causas diretamente relacionadas com a poluição.[1]

Com as emissões excessivas de gases de efeito de estufa mais especificamente Dióxido de Carbono (CO2) é possível observar, para além das consequências na saúde, efeitos negativos relacionados com o meio ambiente como o aquecimento global. O aquecimento global define-se como o processo de aumento da temperatura média dos oceanos e da atmosfera, tendo como exemplo de algumas das consequências, alterações na frequência das chuvas, mais secas e ondas de calor e o aumento do nível das águas, motivado pelo descongelamento dos glaciares do ártico. [2]

[1] Poluição atmosférica - <https://www.sns.gov.pt/noticias/2018/05/02/oms-poluicao-atmosferica/>

[2] <https://climate.nasa.gov/effects/>

**Enquadramento**

O conceito de Carpooling consiste em partilhar lugares disponíveis num veículo próprio. As viagens geralmente estão organizadas em comunidades com interesses comuns, sendo estas pessoas que trabalham na mesma empresa, moram na mesma localidade, que vão para o mesmo evento (cultural, desportivo), entre outros. O Carpooling começou a expandir-se nos estados unidos durante a segunda guerra mundial de forma a conservar recursos necessários.

Este conceito assemelha-se a uma boleia, mas que é requisitada através da internet. Neste projeto será implementado um sistema que permite esta requisição através de um dispositivo móvel.

Como qualquer sistema, este apresenta vantagens e desvantagens. No que toca às vantagens este sistema incentiva a sustentabilidade ambiental, sendo que reduz as emissões de dióxido de carbono emitidas pelos carros, é também uma forma de poupar dinheiro e conhecer pessoas novas da sua comunidade. Entre as desvantagens está o facto de ainda não ser um sistema muito conhecido e como consequência tem menos viagens disponíveis do que numa situação inicial.

**Motivação**

Devido à necessidade e urgência de diminuir os efeitos das alterações climáticas, é fundamental estudar alternativas que irão garantir a vivência das gerações atuais e futuras. Através do Carpooling é possível reduzir os gases com efeito de estufa emitidos pelos carros devido à diminuição de automóveis nas estradas. Não sendo a solução para toda a poluição é um passo em frente na consciencialização da população, tendo diversas vantagens para além da sustentabilidade, sejam estas económicas ou sociais.

Em suma a motivação deste projeto baseou-se em amplificar o conhecimento deste sistema de Carpooling para diminuir a poluição que os transportes privados causam.

**Objetivo**

Atualmente estão disponíveis soluções informáticas que se baseiam essencialmente na gestão de anúncios que referem viagens recorrentes ou de longa distância, ou seja, previamente planeadas. No entanto, o *Carpooling*em zonas urbanas para ser viável exige flexibilizar a subscrição e negociação de anúncios com importância acrescida em viagens esporádicas e de curta duração.

Ao expandir a forma de acesso aos serviços de Carpooling para um modo mais dinâmico e esporádico, haverá um aumento de utilizadores que por sua vez se traduzem em mais viagens disponíveis. No futuro é esperado que este sistema de transporte se transforme numa das formas de transportação sustentável que simultaneamente impulsiona a redução da poluição e o aumento da interação humana.

Neste projeto foi implementada uma solução informática que facilita aos membros da comunidade de *Carpooling* o acesso a informação, em qualquer lugar e em tempo real através de um dispositivo móvel, sobre ofertas de boleias em curso assim como sobre os interessados em aderirem a essas ofertas.

**Plataformas relacionadas**

Como qualquer sistema conhecido, foram desenvolvidas ao longo dos anos várias plataformas que o suportam. Neste caso foram encontradas plataformas semelhantes que englobam o conceito de Carpooling mas que não suportam o lado dinâmico deste projeto de requisitar viagens em tempo real.

Em todas estas plataformas os condutores podem partilhar as informações da viagem que vão realizar e disponibilizar lugares do seu veículo privado e os passageiros podem juntar-se às viagens antes destas ocorrerem. Algumas destas são:

Waze Carpool

O Waze é uma aplicação para dispositivos móveis, baseada na navegação por GPS e é suportada pela Google. Apesar deste sistema ser mais conhecido como um sistema de navegação, tem como uma das suas vertentes o Waze Carpool. Neste sistema os utilizadores condutores colocam um anúncio de uma viagem que vão realizar e os utilizadores passageiros podem se juntar a esta viagem. Apesar do Waze estar disponível em Portugal e ser um auxiliar de condução relativamente conhecido, o Waze Carpool ainda não está disponível.

Waze, “Waze Carpool,” [Online]. Disponível: https://www.waze.com/pt-BR/carpool.

BlaBlaCar

BlaBlaCar é uma das maiores plataformas de Carpooling existentes a nível mundial, foi criada por Francis Nappez e Nicolas Brusson e está disponível em vinte e dois países incluindo Portugal. É descrita como uma rede social que conecta condutores e passageiros e pode ser acedida através da versão Web ou aplicação móvel.

Scoop

Fundado em 2015, o scoop conecta colegas de trabalho e vizinhos ao fornecer um serviço orientado para comunidades como empresas e habitações. Deste modo, utilizadores que forem para a mesma localidade podem-no fazer de forma segura e conveniente.

Tem como parceiras algumas das maiores empresas mundiais como LinkedIn, FedEx e Samsung, reafirmando a sua segurança. E à semelhança do WazeCarpool ainda não está disponível em Portugal.



**Organização do Documento**

Este documento está organizado em quatro capítulos.

O capítulo atual constitui o primeiro capítulo onde é feita uma introdução ao tema deste projeto, um enquadramento do conceito e qual a nossa motivação para o desenvolvimento do mesmo. São também analisadas plataformas semelhantes e há uma descrição geral da estrutura do documento atual.

O segundo capítulo descreve as tecnologias que foram consideradas na implementação deste projeto tal como as vantagens e desvantagens de cada uma e qual a escolhida.

O terceiro capítulo baseia-se na solução informática proposta, onde vão ter detalhadas as tecnologias e arquitetura definida.

**Tecnologias**

Dada a natureza da aplicação a desenvolver, é essencial que esta esteja acessível ao maior número possível de utilizadores. Assim sendo foram analisadas tecnologias que suportam o desenvolvimento para dispositivos móveis, dispositivos estes que operam na sua maioria nos sistemas operativos Android e IOS.

Há então a possibilidade de desenvolver três tipos de aplicações direcionadas para os diferentes sistemas operativos, seguem-se:

**Nativa**

Uma aplicação nativa é desenvolvida para um determinado sistema operativo e numa determinada linguagem de programação. Para IOS as aplicações são desenvolvidas em Objective-C ou swift enquanto para Android são desenvolvidas em Java ou Kotlin.

**Cross-Platform**

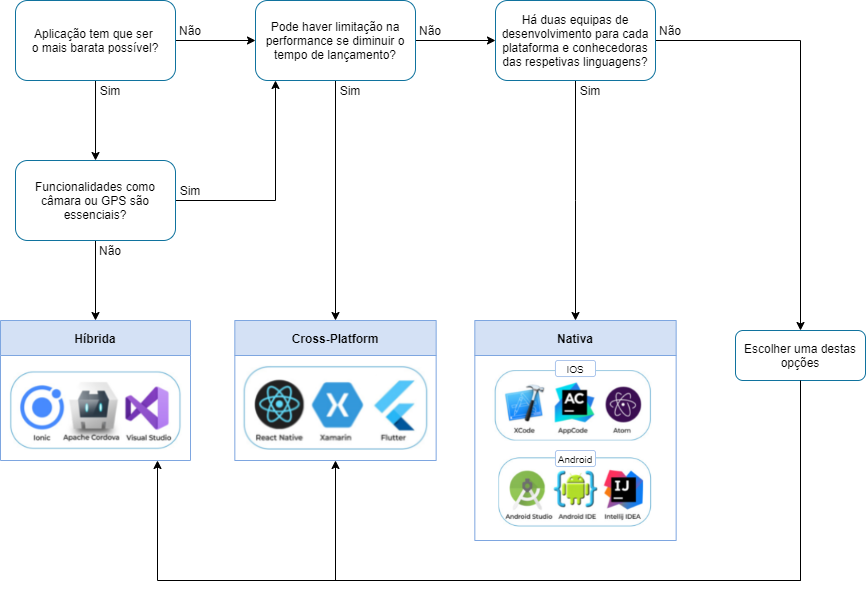
Uma aplicação cross-platform consiste numa aplicação universal que é suportada pelos diferentes sistemas operativos no qual é utilizado apenas um sistema de código. Neste tipo de aplicação há uma experiência de utilizador semelhante à nativa.

**Híbrida**

Híbrido é algo composto por elementos de tipos diferentes. Uma aplicação híbrida é desenvolvida com a mesma tecnologia utilizada na implementação de websites e corre na infraestrutura nativa do dispositivo. É assim uma combinação da parte nativa e das tecnologias Web (HTML5, CSS e JavaScript).

|  |  |  |
| --- | --- | --- |
|  | Vantagens | Desvantagens |
| Nativa | * Melhor performance * Maior flexibilidade * Acesso total a APIs nativas * Maior qualidade | * Código não reutilizável * Não suporta múltiplos sistemas operativos (necessidade de aprender duas linguagens de programação) * Mais custos de desenvolvimento |
| Cross-platform | * Suporta vários sistemas operativos * Elementos de interface e experiência semelhante a nativa * Código reutilizável * Facilidade no desenvolvimento de um produto mais rapidamente e com melhor qualidade | * Dificuldade de integração * Pior performance * Acesso mais restrito a funções do sistema (menos limitado que no híbrido) |
| Híbrida | * Suporta vários sistemas operativos * Manutenção simples * Menos custos de desenvolvimento * Código reutilizável | * Restrição no acesso a funções de sistema operativo * Pior performance * Falta de interação com outras aplicações nativas |

Devido à necessidade do maior número de utilizadores possíveis, uma aplicação nativa não foi considerada como opção viável. De seguida, restam as aplicações cross-platform e híbridas, nas quais foram analisadas as vantagens e desvantagens. Como este projeto requer acesso a funções do sistema como o GPS dos utilizadores, a decisão assentou em desenvolver uma aplicação cross-platform por ser menos limitada.



<https://railsware.com/blog/native-vs-hybrid-vs-cross-platform/>

Dentro do desenvolvimento cross-platform foram encontradas três das frameworks mais conhecidas e utilizadas atualmente. Sendo estas:

**React Native**

É um Framework Javascript desenvolvido pelo Facebook e é utilizado para desenvolver aplicações para os sistemas Android e IOS, assim como para aplicações web. Funciona através de uma *“JavaScript bridge”* que possibilita a comunicação de JavaScript com a linguagem nativa em questão.

**Flutter**

É um *Framework* desenvolvido pelo Google que utiliza a linguagem de programação *Dart* também criada pela empresa. Este facilita a criação de aplicações multi-plataforma com uma interface flexível, mantendo uma performance semelhante à de uma app desenvolvida nativamente.

**Xamarin**

É um Framework desenvolvido pelo Microsoft para desenvolver aplicações para os sistemas Android, IOS e Windows. O Xamarin é uma camada de abstração que gere a comunicação de código partilhado com código de plataforma subjacente.

<https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>

|  |  |  |  |
| --- | --- | --- | --- |
|  | Vantagens | Desvantagens |  |
| React Native | * Aprender uma vez e escrever em qualquer lado * Linguagem Javascript no top 5 de linguagens de programação * Documentação extensa * Grande comunidade de desenvolvedores | * Performance inferior * Dificuldade em fazer Debugging * Complexidade de implementação | [**https://www.thirdrocktechkno**](https://www.thirdrocktechkno)**.**  **com/blog/pros-and-cons-of-react-native-development-in-2021/** |
| Flutter | * Desenvolvimento mais rápido * Melhor performance devido a plugins prontos a usar, widgets e um motor próprio de renderização. * Grande customização | * Recente, acaba por não ter a abrangência das outras frameworks. * Ainda pouco utilizado, mas em crescimento constante de popularidade. * Linguagem dart é relativamente recente e requer aprendizagem | **https://www.futuremind.com/blog/pros-cons-flutter-mobile-development** |
| Xamarin | * Desenvolvimento mais rápido * Performance semelhante à nativa * Manutenção simplificada | * Acesso limitado a bibliotecas open source * Aplicação mais pesada * Problemas de compatibilidade com bibliotecas externas | https://softjourn.com/insights/xamarin-app-development-advantages-and-disadvantages |

Após analisadas as três diferentes frameworks cross-platform, a escolha assentou na linguagem e documentação e comunidade de cada uma. Desta forma o React native foi a melhor opção entre as estipuladas uma vez que é implementada em JavaScript, é uma framework muito completa em termos de documentação sendo complementada por bibliotecas que serão úteis no desenvolvimento deste projeto. Por fim o facto da comunidade ser extensa foi visto como um beneficio

**Proposta**

**Arquitetura**

<https://www.appvelocity.ca/blog/guide-mobile-application-architecture>

A arquitetura de uma aplicação móvel é constituída por três camadas principais, a camada de apresentação, camada de negócio e camada de acesso de dados.

A camada de apresentação é direcionada para os componentes da interface de utilizador. O foco principal desta camada é a apresentação da aplicação ao utilizador final e o design é feito consoante o tipo de utilizador da mesma.

A camada de lógica de negócio é a base da aplicação e é onde são expostas as funcionalidades. É nesta camada que está o servidor da aplicação que comunica com o cliente através de pedidos e respostas.

A Camada de acesso de dados é criada pela combinação de utilidades de dados, componentes de acessos de dados e agentes de serviço, ou seja, é a camada que fornece acesso aos dados armazenados. É nesta camada que é feito o acesso à API fornecida.

Camada de apresentação

Falar dos ecrãs

Camada de lógica de negócio

Ao criar o ambiente do projeto através da framework react native foi necessária a instalação do Node js, que por sua vez, foi reaproveitado para implementar uma relação cliente- servidor na nossa aplicação, servindo de servidor.

Um cliente é a representação de um utilizador, ou neste caso, o dispositivo em que se encontra. Este dispositivo tem a capacidade de receber informação ou utilizar um serviço particular dos servidores. E um servidor é um meio que serve algo e pode ser um programa de computador ou um dispositivo que gere informação a clientes.

Neste projeto o uso de um servidor irá ser essencial pois há a necessidade de troca de coordenadas entre clientes, ou seja, da parte do cliente irá haver um pedido de informação da localização de outro cliente, informação esta que será transmitida através do servidor.

A comunicação entre clientes e servidor pode ser realizada através de WebSockets (WebSocket API). Com esta API é possível enviar mensagens para um servidor e receber respostas orientadas a eventos sem precisar de consultar o servidor para obter uma resposta. Através de websockets há uma comunicação bidirecional, ou seja, tanto o cliente como o servidor podem mandar informação.

https://developer.mozilla.org/en-US/docs/Web/API/WebSockets\_API?retiredLocale=pt-PT

No contexto deste projeto, através do Websocket, o cliente que está a partilhar a sua localização envia uma atualização para o servidor, e o cliente que está a requisitar esta localização, é atualizado pelo servidor.

Para facilitar a implementação de WebSockets, foi utilizada a biblioteca Socket.io que utiliza o protocolo WebSocket e que fornece uma comunicação em tempo real, bidirecional e baseada em eventos entre o cliente e servidor.



Tem como algumas vantagens:

* Fiabilidade (fornece uma alternativa HTTP caso a conexão WebSocket não possa ser restabelecida
* Se a conexão quebra é automaticamente retomada
* Fazer Broadcast de mensagem para todos os clientes ou para um subconjunto de clientes
* Implementação mais acessível

<https://socket.io/docs/v4/index.html>

React native - podemos mostrar mockups ? not sure

Camada de lógica de negócio

Servidor onde vamos tar a mandar as coordenadas e merdas

Acesso ao outro

-- Utilizadores

-- Viagens colocadas

-- detalhe da viagem